ΤΕΙ ΑΘΗΝΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΡΟΝΙΚΗΣ

Όνομα :.....
Επώνυμο :.....
Α.Μ. :.....
Εξάμηνο :.....
Ημερομηνία : 09/02/2012
Εισηγητής : Κουλούρας Γρηγόριος

Α' ΕΞΕΤΑΣΤΙΚΗ ΠΕΡΙΟΔΟΣ ΧΕΙΜΕΡΙΝΟΥ ΕΞΑΜΗΝΟΥ 2011-2012

ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Θέμα 1ο **3.0 μονάδες**

Ο μικροελεγκτής AVR ATmega16 ενσωματώνει ένα «μετατροπέα αναλογικού σήματος σε ψηφιακό» «Analog to Digital Converter» (ADC) με τα παρακάτω χαρακτηριστικά:

α) Technology: Successive Approximation,
β) Resolution: 10 bit,
γ) Accuracy: ±2 bit,
δ) Convertion Rate: 15kSPS,
ε) Input Voltage Range: 0V to +5V,
στ) Input Channels: 8 Multiplexed Single Ended.

Α) Σχεδιάστε σε μπλοκ διάγραμμα και περιγράψτε την λειτουργία ενός μετατροπέα της τεχνολογίας «Διαδοχικών Προσεγγίσεων» ή «Successive Approximation».

(1.5 μονάδες)

Β) Εξηγήστε περιληπτικά τα υπόλοιπα χαρακτηριστικά του παραπάνω ADC.

(0.5 μονάδες)

Γ) Ποιά είναι η διακριτική ικανότητα του παραπάνω ADC και πως υπολογίζεται;

(1.0 μονάδα)

Θέμα 2ο **7.0 μονάδες**

Α) Μικροελεγκτής AVR ATmega16 χρονίζεται στα 8MHz από εξωτερικό κρύσταλλο. Στην πόρτα Β του AVR είναι συνδεδεμένα LED σε συνδεσμολογία θετικής λογικής. Δίνεται το παρακάτω ολοκληρωμένο πρόγραμμα συμβολικής γλώσσας, για τον παραπάνω μικροελεγκτή. Περιγράψτε τι κάνει η κάθε εντολή στα δύο μπλοκ κώδικα που φαίνονται παρακάτω: «Reset Handler» και «Main Loop».

(2.0 μονάδες)

Β) Γράψτε πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον μικροελεγκτή ATmega16, που να ανιχνεύει συνεχώς κλειδί μήκους ενός Byte σε ένα μπλοκ μνήμης SRAM μεγέθους 56 Bytes που περιέχει μη προσημασμένους αριθμούς 8-bit. Το μπλοκ μνήμης αρχίζει από την διεύθυνση $0100 και το κλειδί είναι το $AA. Στο τέλος του ελέγχου το πλήθος των κλειδιών που ανιχνεύτηκαν στο συγκεκριμένο διάστημα μνήμης θα αποθηκεύεται στην διεύθυνση $007A.

(2.5 μονάδες)

Γ) Γράψτε πρόγραμμα σε συμβολική γλώσσα «Assembly» για τον μικροελεγκτή ATmega16, που να υπολογίζει συνεχώς το checksum ενός μπλοκ μνήμης SRAM μεγέθους 200 Bytes που περιέχει προσημασμένους αριθμούς 8-bit. Το μπλοκ μνήμης ξεκινάει από την διεύθυνση $0061. Στο τέλος του υπολογισμού το συμπλήρωμα ως προς δύο του αποτελέσματος θα αποθηκεύεται στην διεύθυνση $0060. (Υπολογισμός του checksum: Το checksum ξεκινάει με την τιμή $FF. Στην συνέχεια κάνοντας διαδοχικά XOR με όλα τα Bytes του Block διαμορφώνεται το τελικό αποτέλεσμα που έχει μέγεθος ένα Byte)

(2.5 μονάδες)

```
;=====================================================================================================;
;Note: Start of Program
;=====================================================================================================;
.include "m16def.inc"
.CSEG
.ORG 0
;=====================================================================================================;
;Note: Interrupt Vectors
;=====================================================================================================;
                jmp RESET               ; Reset Handler
                jmp EXT_INT0            ; IRQ0 Handler
                jmp EXT_INT1            ; IRQ1 Handler
                jmp TIM2_COMP           ; Timer2 Compare Handler
                jmp TIM2_OVF            ; Timer2 Overflow Handler
                jmp TIM1_CAPT           ; Timer1 Capture Handler
                jmp TIM1_COMPA          ; Timer1 CompareA Handler
                jmp TIM1_COMPB          ; Timer1 CompareB Handler
                jmp TIM1_OVF            ; Timer1 Overflow Handler
                jmp TIM0_OVF            ; Timer0 Overflow Handler
                jmp SPI_STC             ; SPI Transfer Complete Handler
                jmp USART_RXC           ; USART RX Complete Handler
                jmp USART_UDRE          ; UDR Empty Handler
                jmp USART_TXC           ; USART TX Complete Handler
                jmp ADC_CONV            ; ADC Conversion Complete Handler
                jmp EE_RDY              ; EEPROM Ready Handler
                jmp ANA_COMP            ; Analog Comparator Handler
                jmp TWSI                ; Two-wire Serial Interface Handler
                jmp EXT_INT2            ; IRQ2 Handler
                jmp TIM0_COMP           ; Timer0 Compare Handler
                jmp SPM_RDY             ;Store Program Memory Ready Handler
;=====================================================================================================;
;Note: Reset Handler
;=====================================================================================================;
RESET:          cli                     ;_____________________________________________________________
                ldi r16, high(RAMEND)   ;_____________________________________________________________
                out SPH,r16             ;_____________________________________________________________
                ldi r16, low(RAMEND)    ;_____________________________________________________________
                out SPL,r16             ;_____________________________________________________________
;=====================================================================================================;
;Note: Main Loop
;=====================================================================================================;
MAIN:           clr r16                 ;_____________________________________________________________
                ser r17                 ;_____________________________________________________________
                out DDRB,r17            ;_____________________________________________________________
                out PORTB,r16           ;_____________________________________________________________
                ldi r27,high($0071)     ;_____________________________________________________________
                ldi r26,low($0071)      ;_____________________________________________________________
                ldi r18,128             ;_____________________________________________________________
LOOP:           ld r17,X                ;_____________________________________________________________
                cpi r17,$80             ;_____________________________________________________________
                brlo SKIP               ;_____________________________________________________________
                andi r17,$7E            ;_____________________________________________________________
                inc r16                 ;_____________________________________________________________
SKIP:           st X+,r17               ;_____________________________________________________________
                dec r18                 ;_____________________________________________________________
                brne LOOP               ;_____________________________________________________________
                out PORTB,r16           ;_____________________________________________________________
                rjmp MAIN               ;_____________________________________________________________
;=====================================================================================================;
;Note: Interrupt Handlers
;=====================================================================================================;
EXT_INT0:       reti                    ; IRQ0 Handler
EXT_INT1:       reti                    ; IRQ1 Handler
TIM2_COMP:      reti                    ; Timer2 Compare Handler
TIM2_OVF:       reti                    ; Timer2 Overflow Handler
TIM1_CAPT:      reti                    ; Timer1 Capture Handler
TIM1_COMPA:     reti                    ; Timer1 CompareA Handler
TIM1_COMPB:     reti                    ; Timer1 CompareB Handler
TIM1_OVF:       reti                    ; Timer1 Overflow Handler
TIM0_OVF:       reti                    ; Timer0 Overflow Handler
SPI_STC:        reti                    ; SPI Transfer Complete Handler
USART_RXC:      reti                    ; USART RX Complete Handler
USART_UDRE:     reti                    ; UDR Empty Handler
USART_TXC:      reti                    ; USART TX Complete Handler
ADC_CONV:       reti                    ; ADC Conversion Complete Interrupt Handler
EE_RDY:         reti                    ; EEPROM Ready Handler
ANA_COMP:       reti                    ; Analog Comparator Handler
TWSI:           reti                    ; Two-wire Serial Interface Handler
EXT_INT2:       reti                    ; IRQ2 Handler
TIM0_COMP:      reti                    ; Timer0 Compare Handler
SPM_RDY:        reti                    ; Store Program Memory Ready Handler
;=====================================================================================================;
;Note: End of Program
;=====================================================================================================;
```